

Joint Rain Detection and Removal via Iterative Region Dependent Multi-Task Learning

Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, Shuicheng Yan

Abstract—In this paper, we address a rain removal problem from a single image, even in the presence of heavy rain and rain accumulation. Our core ideas lie in our new rain image models and a novel deep learning architecture. We first modify the commonly used model, which is a linear combination of a rain streak layer and a background layer, by adding a binary map that locates rain streak regions. Second, we create a model consisting of a component representing rain accumulation (where individual streaks cannot be seen, and thus visually similar to mist or fog), and another component representing various shapes and directions of overlapping rain streaks, which normally happen in heavy rain. Based on the first model, we develop a multi-task deep learning architecture that learns the binary rain streak map, the appearance of rain streaks, and the clean background, which is our ultimate output. The additional binary map is critically beneficial, since its loss function can provide additional strong information to the network. In many cases though, rain streaks can be dense and large in their size, thus to obtain the clean background, we need spatial contextual information. For this, we utilize the dilated convolution. To handle rain accumulation (again, a phenomenon visually similar to mist or fog) and various shapes and directions of overlapping rain streaks, we propose an iterative information feedback (IIF) network that removes rain streaks and clears up the rain accumulation iteratively and progressively. Overall, this multi-task learning and iterative information feedback benefits each other and constitutes a network that is end-to-end trainable. Our extensive evaluation on real images, particularly on heavy rain, shows the effectiveness of our novel models and architecture, outperforming the state-of-the-art methods significantly.

Index Terms—Rain Removal, Rain Detection, Multi-Task Learning, Region Dependent Model, Iterative Information Feedback

1 INTRODUCTION

RESTORING rain images is important for many computer vision applications in outdoor scenes. Rain degrades visibility significantly and causes many computer vision systems to likely fail. Generally, rain introduces a few types of visibility degradation. Raindrops obstruct, deform and/or blur the background scenes. Distant rain streaks accumulate and generate atmospheric veiling effects similar to mist or fog, which severely reduce the visibility by scattering light out and into the line of sight. Nearby rain streaks exhibit strong specular highlights that occlude background scenes. These rain streaks can have various shapes and directions, particularly in heavy rain, causing severe visibility degradation.

In the past decades, many researchers have devoted their attention to solving the problem of restoring rain images. The proposed methods can be categorized into two classes. One class regards the rain streak removal problem as a signal separation (or image layer decomposition) problem. A series of approaches have been proposed including: the morphological component analysis with sparse coding [21], [27], structural similarity constraints [38], generalized low rank model [9] and discriminative sparse coding [33]. The other class is filter-based or diffusion-based method, e.g. the nonlocal mean smooth [29].

While there are varying degrees of success, the majority of existing methods suffer from several limitations:

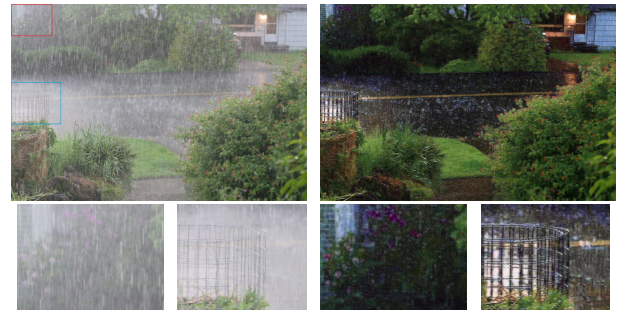


Fig. 1. An example of our rain removal method to both remove rain streaks and enhance the visibility, achieving considerably promising results. left: rain image, right: processed result, top: whole image, down: local regions.

- Due to the intrinsic overlapping between rain streaks and background texture patterns, most methods tend to remove texture details in non-rain regions, leading to over-smoothing the regions.
- The degradation of rains is complex, and the existing image formation model [9], [27] widely used in previous methods is insufficient to cover some important factors in real rain images, such as the atmospheric veils due to rain accumulation, and different shapes or directions of streaks.
- The basic operation of many existing algorithms is on a local image patch or a limited receptive field (a limited spatial range). Thus, spatial contextual information in larger regions, which has been proven to be useful for rain removal [22], is rarely used.

Considering these limitations, our goal is to develop a novel rain image formation model that is more capable of describing various

Wenhan Yang, Jiaying Liu and Zongming Guo are with Institute of Computer Science and Technology, Peking University, Beijing, 100080, P.R. China, e-mail: {yangwenhan, liujiaying, guozongming}@pku.edu.cn.

Jiashi Feng and Shuicheng Yan are with Department of Electrical and Computer Engineering, National University of Singapore, Singapore, e-mail: {elefjia, eleyans}@nus.edu.sg.

Robby T. Tan is with both Yale-NUS College and Department of Electrical and Computer Engineering, National University of Singapore, Singapore, e-mail: robby.tan@yale-nus.edu.sg.

rain conditions in real scenes, including rain accumulation and heavy rain, and then, use them to design an effective deep learning architecture. Here, we focus on a single input image.

To achieve the goal, we explore the possible rain models and deep learning architectures that can effectively restore rain images even in the presence of heavy rain. Our ideas are as follows. First, we introduce novel region-dependent rain image formation models that can capture the nature of rain images better. In the models, we use a rain-streak binary map, where 1 indicates the presence of individually visible rain streaks in the pixels, and 0 otherwise. We also model the appearance of rain accumulation (which is similar to that of mist or fog), and the various shapes and directions of overlapping streaks, to simulate heavy rain.

Second, based on our introduced models, we construct a deep network that jointly detects and removes rain. Rain streak regions are detected by the network and it is used to constrain the rain removal. With this, our network is capable of performing an adaptive operation on rain and non-rain regions, preserving richer details.

Third, to retrieve more contextual information, we propose a multi-scale aggregated recurrent residual network (MSA-ResNet) to enlarge the receptive field. In this network, the extracted features are enhanced in each recurrence progressively by the aggregated information from several parallel dilated convolutions.

Finally, to restore images captured in the environment with both rain accumulation and various rain streak directions, we propose an iterative information feedback (IIF) network that progressively removes rain streaks. Extensive experiments and evaluations demonstrate that our method outperforms state-of-the-art methods significantly on both synthesized data and real data. Particularly for some heavy rain images, our method still achieves considerably good results.

Hence, our contributions are:

- 1) The first method to model the rain-streak binary mask, and also to model the atmospheric veils due to rain accumulation as well as various shapes and directions of overlapping rain streaks. This enables us to synthesize data for training more similar to real rain images.
- 2) The first method to jointly detect and remove rains from single images. With the additional information from rain region detection, the further rain removal approach achieves better performance.
- 3) The first rain removal method that uses a multi-scale recurrent residual network to obtain more context while preserving rich local details.
- 4) The first method that addresses heavy rain by introducing an iterative information feedback network, where it removes rain progressively, enabling us to obtain good results even in significantly complex cases.

The rest of this paper is organized as follows. Section 2 gives a brief review on existing video and single image rain removal methods, as well as some recent proposed DL-based image processing methods. Section 3 is devoted to our new proposed region-dependent rain image model. Section 4 constructs a multi-task learning network to perform joint rain detection and removal (JORDER) to solve the inverse problem raised in Section 3. Section 5 illustrates the extension of JORDER to handle rains in real cases with iterative information feedback and an embedded dehazing network. Experiments are presented in

Section 6 and deeper analysis is provided in Section 7. Concluding remarks are given in Section 8.

2 RELATED WORK

Rain removal from videos Rain image recovery [2]–[4], [9], [13], [15], [17], [19], [48] from video sequences has been widely explored. Garg *et al.* [15]–[17], [19] first constructed the appearance model to describe rain streaks and exploited it to detect rain pixels in video. Zhang *et al.* [48] and Brewer *et al.* [4] focused on the chromaticity and shape of rain streaks, respectively. Other methods construct novel features to model and detect rain streaks, such as frequency domain analysis [2], histogram of orientation [3] and generalized low rank [9]. These methods make full use of the rich information in videos and the temporal redundancy in adjacent frames to identify rain streaks. In contrast, our method attempts to jointly detect and remove rain regions from only a single image.

Rain removal from single image Compared with the multi-frame removal problem, without the temporal information, the recovery from a single rain image is more ill-posed. Some single-image based rain removal methods regards the problem as a layer separation problem. Huang *et al.* [27] attempted to separate the rain streaks from the high frequency layer by sparse coding, with a learned dictionary from the HOG features. However, the capacity of morphological component analysis, the layer separation, and learned dictionary is limited. Thus, it usually causes the over-smoothness of the background. In [9], a generalized low rank model is proposed, where the rain streak layer is assumed to be low rank. In [28], Kim *et al.* first detected rain streaks and then remove them with the nonlocal mean filter. In [33], Luo *et al.* proposed a discriminative sparse coding method to separate rain streaks from background images. A recent work of [30] exploit the Gaussian mixture models to separate the rain streaks, achieving the state-of-the-art performance, however, still with slightly smooth background. In this paper, we use the deep network to perform joint rain detection and removal, with the priors and constraints learned automatically from the synthesized data.

Deep learning image processing In recent years, deep learning-based image processing applications emerged with promising performance. These applications include denoising [1], [5], [6], [25], [39], completion [42], super-resolution [10]–[12], [35], deblurring [37], deconvolution [43] and style transfer [20], [44], *etc.* There are also some recent works on bad weather restoration or image enhancement, such as dehazing [7], [41], rain drop and dirt removal [13] and light enhancement [32]. Besides, with the superior modeling capacity than shallow models, DL-based methods begin to solve harder problems, such as blind image denoising [47]. In this paper, we exploit the deep network to jointly detect and remove rain.

3 REGION-DEPENDENT RAIN IMAGE MODEL

We briefly review the existing rain image formation model, and extend it to explicitly include a rain-streak binary map. We also introduce a novel rain formation model that captures rain accumulation (atmospheric veils) and rain streaks that have various shapes and directions.

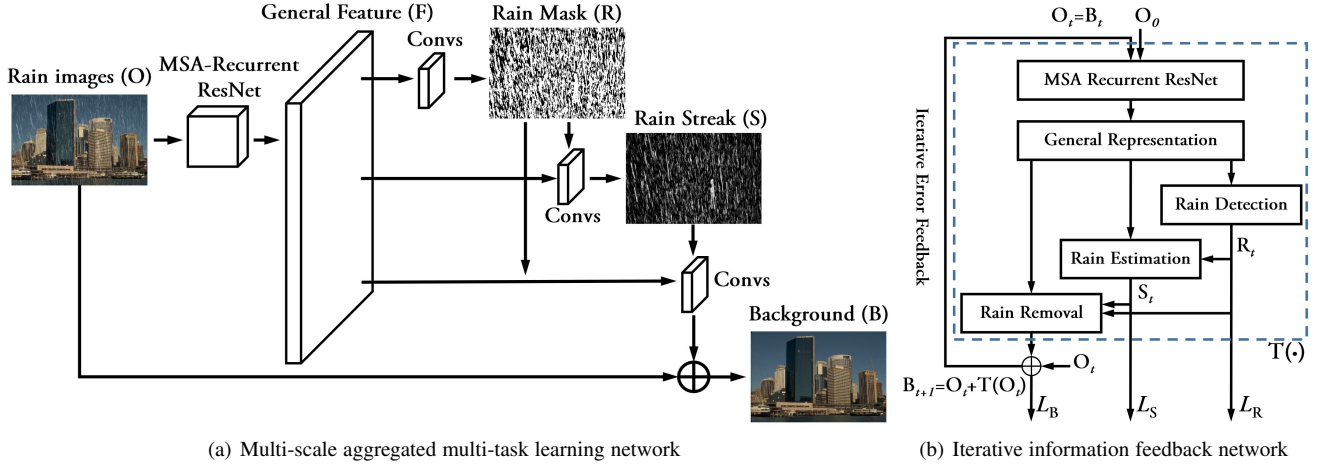


Fig. 2. The architecture of our proposed multi-scale aggregated multi-task network (a), and cascaded iterative multi-task information feedback network (b). (a) A multi-scale aggregated recurrent ResNet extracts general features F from the input rain image O . Then, R , S , B and are predicted in a sequential order. Each variable is predicted based on F and the predicted variables from previous steps. (b) In order to handle practical rain images including heavy rains, we further construct an iterative information feedback network by cascading several multi-scale aggregated multi-task learning networks to removal rains progressively.

3.1 Region-Dependent Rain Image Formation

The widely used rain image formation model is expressed as:

$$O = B + \tilde{S}, \quad (1)$$

where B is the background scene without the effect of rain streaks, and \tilde{S} is the rain streak layer. O is the captured image with rain streaks. Based on this model, deraining is regarded as an image signal separation problem, namely that, given the observation O , estimate the background B and rain streak \tilde{S} , based on the intrinsic differences between the natural images and rain streaks, such as their texture patterns [27], [30] or the group directional properties of rain streaks [31]. The methods that deeply root in Eq. (1) have two deficiencies. First, \tilde{S} is a non-uniformly sparse map, which is dense for heavy rain regions but sparse for rainless ones, thus it is hard to model \tilde{S} with existing sparsity priors with a fixed sparse level. Second, solving a pure signal separation problem Eq. (1) naturally leads to a non-distinctive process on both heavy rain and rainless regions, thus over-smoothing rainless regions.

To overcome these drawbacks, we extend the rain image formation model as follows,

$$O = B + SR, \quad (2)$$

where a new region-dependent variable R is introduced to indicate the existence of individually visible rain streaks. R is a binary rain mask, where 1 indicates rain regions, whereas 0 indicates non-rain regions. Note that, although R is easily estimated by an indicator function on S , modeling R separately from S provides two desirable capabilities for a deep learning network: 1) it provides additional information for the network to learn detecting rain streak regions, 2) it enables to construct a joint pipeline that detects rain regions first, and then operate discriminatively on rain-streak and non-rain-streak regions, thus richer background details could be preserved.

3.2 Rain Accumulation and Heavy Rain

In real world, rain is not only about rain streaks that are individually observable. Depending on the density and distance, rain is also



Fig. 3. In heavy rain cases, the rain streaks have various shapes and directions (shown in blue windows). Rain accumulation reduces the visibility for distance scenes (shown in red windows).

formed by rain streak accumulation, where the individual streaks can be no longer seen. Rain accumulation, which visually behaves like mist or fog, causes the atmospheric veiling effect and also blur, particularly for distance scenes, as shown in Figure 3 (a). Aside from rain accumulation, rain is not only composed of rain streaks that have the same direction. In many occasions, particularly in heavy rain, the rain streaks have various shapes and directions that overlap to each other, as shown in Figure 3 (a) and 3 (b).

For these two rain phenomena, we further extend Eq. (2) to create a new model to accommodate them:

$$O = \alpha \left(B + \sum_{t=1}^s \tilde{S}_t \right) + (1 - \alpha) A, \quad (3)$$

where each \tilde{S}_t is the direction- and shape-consistent individually-visible rain streaks, t is the overlapping streak number, and s is the number of shape and direction consistent rain streaks within an image. A is the global atmospheric light, α is the scene transmission. Based on Eq. (3), we could synthesize the rain accumulation and heavy rain images as the training data, which is closer to the appearance of natural rains, as shown in Figure 4. Note that the atmospheric veiling effect is enforced on the rain-contaminated image $(B + \sum_{t=1}^s \tilde{S}_t)$, thus Eq. (3)



Fig. 4. Synthesized rain images following the process of Eq. (3). Both rain accumulation and multiple rain streaks overlapping are considered. Top: original RGB images from Make3D [36] dataset, Down: our synthesized rain images based on RGB-D data.

implies that, we can handle rain accumulation and rain streak removal separately.

4 DEEP JOINT RAIN DETECTION AND REMOVAL

We construct a multi-task learning deep network to perform joint rain detection and removal (JORDER) to solve the inverse problem raised by Eq. (2). Rain regions are first detected by JORDER, which provides useful side-information to further constrain the rain removal. To percept more context information while preserving richer local details, our network is augmented by multi-scale aggregations via dilated convolution.

4.1 Multi-Task Learning for Joint Rain Detection and Removal

Relying on Eq. (2), given the observed rain image \mathbf{O} , our goal is to estimate \mathbf{B} , \mathbf{S} and \mathbf{R} . Due to the ill-posedness nature of the problem, it leads to a maximum-a-posteriori (MAP) estimation:

$$\arg \min_{\mathbf{B}, \mathbf{S}, \mathbf{R}} \|\mathbf{O} - \mathbf{B} - \mathbf{S}\mathbf{R}\|_2^2 + P_b(\mathbf{B}) + P_s(\mathbf{S}) + P_r(\mathbf{R}), \quad (4)$$

where $P_b(\mathbf{B})$, $P_s(\mathbf{S})$ and $P_r(\mathbf{R})$ are the enforced priors on \mathbf{B} , \mathbf{S} , \mathbf{R} respectively. Previous priors on \mathbf{B} and \mathbf{S} include hand-crafted features, *e.g.* cartoon texture decomposition [27], and some data-driven models, such as sparse dictionary [33] and Gaussian mixture model [30]. For deep learning methods, the priors about

\mathbf{B} , \mathbf{S} and \mathbf{R} are learned from the training data and are embedded into the network implicitly.

The estimation of \mathbf{B} , \mathbf{S} and \mathbf{R} is intrinsically correlated. Thus, the estimation of \mathbf{B} benefits from the predicted $\hat{\mathbf{S}}$ and $\hat{\mathbf{R}}$. To solve this, the natural choice is to employ a multi-task learning architecture, which can be trained using multiple loss functions based on the ground truths of \mathbf{R} , \mathbf{S} and \mathbf{B} , as shown in Figure 2. We first exploit a multi-scale aggregated recurrent residual network to extract the general feature representation \mathbf{F} . Subsequently, \mathbf{R} , \mathbf{S} and \mathbf{B} are predicted in a sequential order, implying a continuous process of rain streak detection, estimation and removal. Each of them is predicted based on \mathbf{F} :

- 1) \mathbf{R} is estimated by two convolutions on \mathbf{F} ,
- 2) \mathbf{S} is predicted by a convolution on the concatenation $[\mathbf{F}, \hat{\mathbf{R}}]$,
- 3) \mathbf{B} is computed from a convolution on the concatenation $[\mathbf{F}, \hat{\mathbf{R}}, \hat{\mathbf{S}}, \mathbf{O} - \hat{\mathbf{R}}\hat{\mathbf{S}}]$.

In fact, there are several potential choices for the network structure design, such as estimating in an order of \mathbf{S} , \mathbf{R} and \mathbf{B} , or in parallel (predicting \mathbf{S} and \mathbf{R} simultaneously). We compare some alternative architectures and demonstrate the superiority of ours empirically in Section 7.2.

4.2 Multi-Scale Aggregated Recurrent ResNet

The recurrent ResNet proposed in [45] has been proven to be effective for low-level image processing tasks. Thus, we utilize it to extract general features. It first transforms the input image into feature space, then enhances the features progressively, and finally obtains the enhanced features \mathbf{F} .

For rain removal tasks, contextual information is demonstrated to be useful for automatically identifying rain patterns and removing rain streaks [23]. Thus, we extend the network structure of [45] with the multi-path recurrent structure, to enlarge the receptive field and obtain contextual information from a larger region. Specifically, in each recurrence, the results from three convolution paths with different dilated factors are aggregated. The dilated convolution [46] weights pixels with a step size called dilated factor, and thus increases its receptive field without sacrificing resolution. For example, a 3×3 dilated convolution with a dilated factor 2 has a receptive field 5×5 . Thus, as shown in Figure 5, the path that perceives the remotest range in each recurrence has the receptive fields up to 9×9 .

To provide a formal description, let \mathbf{f}_{in}^k denote the input feature map for the recurrent subnetwork at the k -th time step. The output feature map \mathbf{f}_{out}^k of the recurrent subnetwork is progressively updated as follows:

$$\mathbf{f}_{out}^k = \max \left(0, \sum_{t=1}^3 (\mathbf{W}_{mid,t}^k * \mathbf{f}_{mid,t}^k + \mathbf{b}_{mid,t}^k) \right) + \mathbf{f}_{in}^k, \quad (5)$$

with $\mathbf{f}_{mid,t}^k = \max (0, \mathbf{W}_{in}^k * \mathbf{f}_{in,t}^k + \mathbf{b}_{in,t}^k),$

where $\mathbf{f}_{in}^k = \mathbf{f}_{out}^{k-1}$ is the output features by the recurrent subnetwork at $(k-1)$ -th time step, and the iteration variable t denotes the sequence number of dilated convolution paths. Note that, the by-pass connection here is between \mathbf{f}_{in}^k and \mathbf{f}_{out}^k . The feature map \mathbf{f}_{out}^k can be viewed as the recovered k -th layer details of the feature maps. Denoting K as the total recurrence number of the

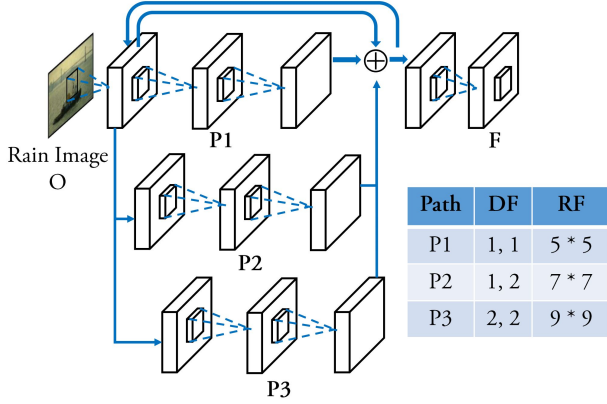


Fig. 5. Multi-scale aggregated recurrent residual network. During the progressive recovery in feature space, for each recurrence, the results from three convolution paths with different dilated factors (DF) and receptive fields (RF) are aggregated.

sub-networks, then the relation between \mathbf{f}_{in}^1 , \mathbf{f}_{out}^K and the overall network is

$$\begin{aligned} \mathbf{f}_{in}^1 &= \max(0, \mathbf{W}_{input} * \mathbf{f}_{input} + \mathbf{b}_{input}), \\ \mathbf{F} &= \mathbf{f}_{out}^K, \end{aligned} \quad (6)$$

where \mathbf{W}_{input} and \mathbf{b}_{input} denote the filter parameter and basis of the convolution layer before the recurrent subnetwork. Hence, \mathbf{F} is the output general features of the multi-scale aggregated recurrent ResNet, as one of the inputs of successive subnetworks.

4.3 Loss Functions

Let $\mathbf{F}_{rr}(\cdot)$, $\mathbf{F}_{rs}(\cdot)$ and $\mathbf{F}_{bg}(\cdot)$ denote the inverse recovery functions modeled by the learned network to generate the estimated rain streak binary map $\hat{\mathbf{R}}$, rain streak map $\hat{\mathbf{S}}$ and background image $\hat{\mathbf{B}}$ based on the input rain image \mathbf{O} . We use Θ to collectively denote all the parameters of the network:

$$\Theta = \{\mathbf{W}_{input}, \mathbf{b}_{input}, \mathbf{W}_{in}, \mathbf{b}_{in}, \mathbf{W}_{mid}, \mathbf{b}_{mid}, \mathbf{W}_{rr}, \mathbf{b}_{rr}, \mathbf{W}_{rs}, \mathbf{b}_{rs}, \mathbf{W}_{bg}, \mathbf{b}_{bg}\}. \quad (7)$$

We use n sets of corresponding rain images, background images, rain region map and rain streak maps $\{(\mathbf{o}_i, \mathbf{g}_i, \mathbf{r}_i, \mathbf{s}_i)\}_{i=1}^n$ for training. We adopt the following joint loss to train the network parameterized by Θ such that it is capable to jointly estimate \mathbf{r}_i , \mathbf{s}_i and \mathbf{g}_i based on rain image \mathbf{o}_i :

$$\begin{aligned} L(\Theta) &= \frac{1}{n} \sum_{i=1}^n (\|\mathbf{F}_{rs}(\mathbf{o}_i; \Theta) - \mathbf{s}_i\|^2 + \lambda_1 \|\mathbf{F}_{bg}(\mathbf{o}_i; \Theta) - \mathbf{g}_i\|^2 \\ &\quad - \lambda_2 (\log \hat{\mathbf{r}}_{i,1} \mathbf{r}_{i,1} + \log(1 - \hat{\mathbf{r}}_{i,2})(1 - \mathbf{r}_{i,2}))), \\ &\text{with } \hat{\mathbf{r}}_{i,j} = \frac{\exp\{\mathbf{F}_{rs,j}(\mathbf{o}_i; \Theta)\}}{\sum_{k=1}^2 \exp\{\mathbf{F}_{rs,k}(\mathbf{o}_i; \Theta)\}}, j \in \{1, 2\}. \end{aligned} \quad (8)$$

Here λ_1 and λ_2 are the parameters to balance the importance among estimating \mathbf{S} , \mathbf{R} and \mathbf{B} .

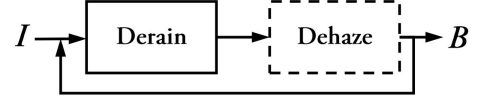


Fig. 6. The schematic diagram of the alternative joint derain and dehaze. The dehazing is only performed once in the first iteration.

5 HANDLING REAL RAIN IMAGES

In the previous section, we constructed a multi-scale aggregated mask-task learning network to jointly detect, estimate and remove rain streaks from rain images. In this section, we enhance our network to handle rain accumulation and rain streaks that possibly have various shapes and directions in one image.

5.1 Iterative Information Feedback Network

The main idea of our solution is to cascade a few of our proposed multi-scale aggregated mask-task learning networks to construct an iterative information feedback (IIF) network, which removes rain, part by part.

Formulation We define the process of the network in the blue dash box of Figure 2 (b) that generates the residual image between \mathbf{O} and \mathbf{B} as $\mathbf{T}(\cdot)$. Then, IIF works as follows,

$$\begin{aligned} \epsilon_t &= \mathbf{T}(\mathbf{O}_t), \\ \mathbf{B}_t &= \mathbf{O}_t + \epsilon_t, \\ \mathbf{O}_{t+1} &= \mathbf{B}_t, \end{aligned} \quad (9)$$

In each iteration t , the predicted residue ϵ_t is accumulated and propagated to the final estimation via updating \mathbf{O}_t and \mathbf{B}_t . The final estimation can be expressed as:

$$\mathbf{B}_\tau = \mathbf{O}_0 + \sum_{t=1}^{\tau} \epsilon_t, \quad (10)$$

where τ is the total iteration number. Hence, the process removes the rain streak progressively, part by part, based on the intermediate results from the previous step. The complexity of rain removal in each iteration is consequently reduced, enabling better estimation.

Loss Functions We extend the loss $L(\Theta)$ in Eq. (8) to $L(\Theta_t, t)$ with t to denote the iteration number, where $L(\Theta_0, 0) = L(\Theta_0)$ in Eq. (8). When $t > 1$, $L(\Theta_t, t)$ equals to the version of $L(\Theta)$ that replaces \mathbf{o}_i and Θ by $\mathbf{o}_{i,t}$ and Θ_t , respectively, where $\mathbf{o}_{i,t}$ is generated from t iterations of the process Eq. (9) on the initial \mathbf{o}_i . Then, the total loss L_{Iter} for training \mathbf{T} is

$$L_{\text{Iter}}(\{\Theta_0, \dots, \Theta_\tau\}) = \sum_{t=0}^{\tau} L(\Theta_t, t), \quad (11)$$

5.2 Alternative Joint Derain and Dehaze

Distant rain streaks accumulate and reduce the visibility by creating an atmospheric veil. To handle this, it is necessary to clear up the atmospheric veil, which is a similar problem to dehazing or defogging.

Considering Eq. (3), it seems to be reasonable for joint deraining and dehazing to start from dehazing to estimate $\mathbf{B} + \sum_{t=1}^s \hat{\mathbf{S}}_t$ first. However, placing dehazing as a preprocessing has complicated effects on deraining, since all rain streaks (including the ones that are already sharp and clearly visible) are boosted, making the rain streaks look different from those in the training images.

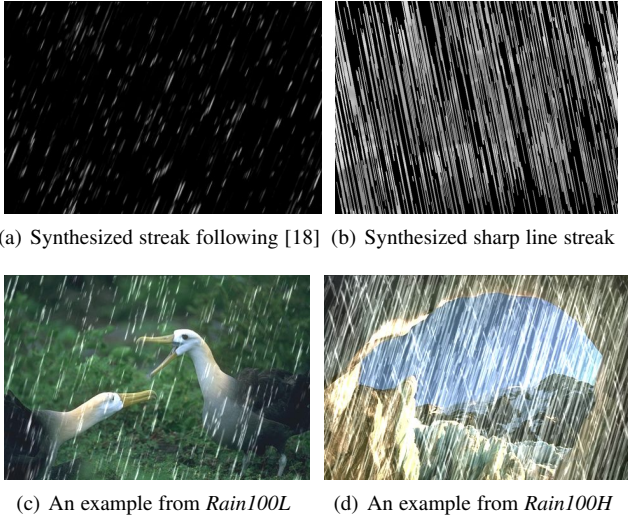


Fig. 7. The examples of our synthesized rain streaks and rain images.

Hence, in our method, we derain first followed by dehazing. This, as it turns out, is beneficial, since dehazing will make the appearance of less obvious rain streaks (which are likely unnoticed by the first round of deraining) become more obvious.

We implement a dehazing network based on the structure of the multi-scale recurrent ResNet, with only one recurrence, trained with the synthesized data generated with the random background reliance and transmission value [8]. Furthermore, we evaluate several possible combinations of joint derain and dehaze empirically in different orders in Section 7.3. The conclusion is that a three-step operation in a sequential order: derain-dehaze-derain is generally effective. Some obvious rain streaks, noises and artifacts are removed in the first round deraining. Subsequently, the dehazing cleans up the rain accumulation, enhances the contrast and visibility, and at the same time boosts weak rain streaks. The second round of the deraining removes these boosted rain streaks, as well as artifacts caused by dehazing, making the results cleaner.

6 EXPERIMENTAL RESULTS

Datasets We compare our method with state-of-the-art methods on four benchmark datasets. First, *Rain12*¹ [30] *Rain100L*, which is our synthesized data set with only one streak direction (Fig. 7 (c)). Second, *Rain20L*, which is a subset of *Rain100L* and used for testing. Third, *Rain100H*, which is our synthesized data set with five streak directions (Fig. 7 (d)). Note, while it is rare for a real rain image to contain rain streaks in many different directions, synthesizing this kind of images as the training set boosts the capacity of the network.

The 100 images for synthesizing *Rain100L*, *Rain20L* and *Rain100H* are selected from BSD200 [34]. The dataset for training our network and another deep learning based baseline – SRCNN for deraining – is BSD300 [34], excluding the ones appeared in *Rain12*. The rain streaks are synthesized in two ways: 1) the photorealistic rendering techniques proposed by [18] as shown in Figure 7 (a); 2) the simulated sharp line streaks along a certain direction with a small variation within an image as shown in Figure 7 (b).

1. <http://yu-li.github.io/>

Baseline Methods We compare four versions of our approaches: JORDER (Section 4), JORDER-IIF (Section 5.1), JORDER-IIF-DEHAZE (Section 5.2) with five state-of-the-art methods: image decomposition (ID) [27], CNN-based rain drop removal (CNN) [14], discriminative sparse coding (DSC) [33], layer priors (LP) [30] and a common CNN baseline for image processing – SRCNN [26], trained for deraining.

SRCNN is implemented and trained by ourselves, while other methods are kindly provided by the authors. SRCNN was trained on Caffe platform [26] via stochastic gradient descent (SGD) with standard backpropagation. We set the momentum as 0.9, the learning rate as a fixed value 10^{-4} for front-end layers and 10^{-5} for the penultimate layer during the training. We allow at most 5×10^7 backpropagations, namely 2.2×10^5 epochs.

Evaluation Metrics For the experiments on synthesized data, two metrics Peak Signal-to-Noise Ratio (PSNR) [24] and Structure Similarity Index (SSIM) [40] are used as comparison criteria. We evaluate the results only in the luminance channel, which has a significant impact on the human visual system to perceive the image quality.

6.1 Quantitative Evaluation

Table 1 and 2 show the results of different methods on *Rain12*. As observed, our method considerably outperforms other methods in terms of both PSNR and SSIM, even another well known deep learning baseline – SRCNN, with a performance gain more than 1.5dB.

Table 3 presents the results of JORDER and JORDER-IIF on *Rain100H*. We do not compare other previous methods because they are not designed for handling such hard cases. It is observed that, IIF significantly boosts the performance on synthesized heavy rain images. The PSNR results of JORDER-IIF gains over JORDER more than 1dB.

TABLE 3
PSNR and SSIM results of JORDER and JORDER-IIF on *Rain100H*.

Metric	JORDER	JORDER-IIF
PSNR	22.15	23.45
SSIM	0.6735	0.7490

6.2 Qualitative Evaluation

We also compare different methods qualitatively on *Rain12*. One can observe that, ID generates smooth results and LP smooths the results slightly. DSC preserves details better and is good at removing large rain occlusion as in *Image5*, but it fails to remove many small rain streaks. Especially, it causes severe artifacts in *Image8* where a white cloud is nearby. Our JORDER produces the cleanest results with the fewest detail loss.

The results of JORDER and JORDER-IIF on *Rain100H* are presented in Figure 9. It is clearly demonstrated that, IIF removes rain streak more effectively while keeps richer details.

6.3 Evaluation on Real Images

Figure 10 presents the results using real images. For fair comparisons, we use JORDER-IIF to process these rain images and do not handle the atmospheric veils on these results, to keep consistent with other methods. It is obvious that, our method significantly outperforms them and is successful in removing the majority of rain streaks.

TABLE 1
PSNR results among different rain streak removal methods on Rain12.

Baseline	1	2	3	4	5	6	7	8	9	10	11	12	Average
ID	28.58	28.50	25.35	30.07	24.96	25.63	27.47	27.40	27.79	27.44	25.84	27.52	27.21
DSC	30.74	30.38	25.77	36.12	27.93	29.15	33.04	27.66	32.22	28.74	28.91	29.60	30.02
LP	31.54	33.31	28.06	36.82	29.34	34.86	35.67	30.95	32.73	30.74	29.02	31.20	32.02
CNN	27.60	26.73	23.26	31.01	25.01	25.59	27.80	24.81	27.90	26.83	26.75	26.53	26.65
SRCNN	33.77	35.19	29.50	40.07	31.33	35.09	39.16	34.92	35.71	32.78	31.44	33.89	34.41
JORDER	34.76	37.45	31.19	43.00	33.51	36.38	41.45	36.94	36.94	33.97	31.98	34.62	36.02

TABLE 2
SSIM results among different rain streak removal methods on Rain12.

Baseline	1	2	3	4	5	6	7	8	9	10	11	12	Average
ID	0.74	0.79	0.84	0.77	0.63	0.73	0.82	0.77	0.74	0.74	0.65	0.77	0.75
DSC	0.83	0.88	0.76	0.96	0.92	0.93	0.94	0.81	0.90	0.82	0.85	0.80	0.87
LP	0.89	0.93	0.92	0.94	0.90	0.95	0.96	0.90	0.91	0.90	0.86	0.92	0.91
CNN	0.75	0.79	0.71	0.89	0.76	0.80	0.85	0.77	0.81	0.76	0.79	0.73	0.78
SRCNN	0.92	0.95	0.90	0.98	0.95	0.97	0.98	0.93	0.96	0.92	0.91	0.93	0.94
JORDER	0.93	0.97	0.93	0.99	0.96	0.98	0.99	0.96	0.97	0.94	0.92	0.94	0.96

6.4 Extreme Cases

We compare all the methods in two extreme cases: dense rain accumulation, and heavy rain. The results are presented in Figure 11. Our method achieves promising results in removing the majority of rain streaks, enhancing the visibility and preserving details.

6.5 Running Time

Table 4 compares the running time of several state-of-the-art methods. All baseline methods and our method are implemented in MATLAB. CNN rain drop and some versions of our method are implemented based on GPU (denoted with “GPU”) while others are based on CPU. It is observed that, our GPU versions run very quickly and the CPU version of JORDER, a lightest version of our method, takes up the shortest running time among all CPU-based approaches. In general, our method is capable of dealing with a 500×500 rain image within 10s and some lighter versions are faster, which is promising for the practical requirement.

7 FURTHER ANALYSES

7.1 Comparison with Potential Network Structures

We compare the proposed framework in Figure 2 with other potential network structures that also jointly estimate $\mathbf{B}, \mathbf{S}, \mathbf{R}$. From the analysis in Section 4.1 based on Eq.(2), generally we have two choices for the network structures as shown in Figure 12: parallel and sequential. Specifically, considering the prediction order of the variables, there are three candidates:

- 1) The parallel structure (predicting \mathbf{S} and \mathbf{R} based on \mathbf{F}),
- 2) The sequential structure (predicting \mathbf{R}, \mathbf{S} and \mathbf{B} in order), denoted as **RSB**, which is our final proposed architecture,
- 3) The sequential structure (predicting \mathbf{S}, \mathbf{R} and \mathbf{B} in order), denoted as **SRB**.

Besides, we also compare with two vanilla versions: 4) a raw four-layer ResNet with only one recurrence to directly predict the background image, denoted as **RAW**, and 5) a four-layer ResNet that uses parts of the features in the penultimate layer to predict \mathbf{S} , denoted as **STR**. Note, all the experiments here do not exploit the multi-scale aggregation.

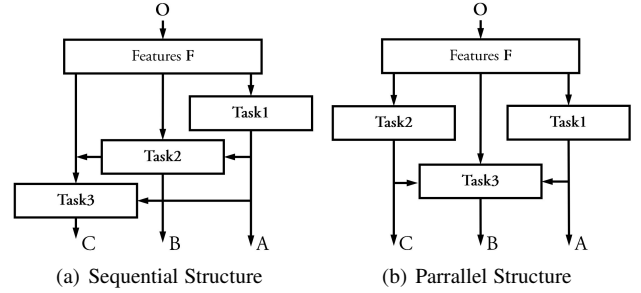


Fig. 12. Potential choices for network structures.

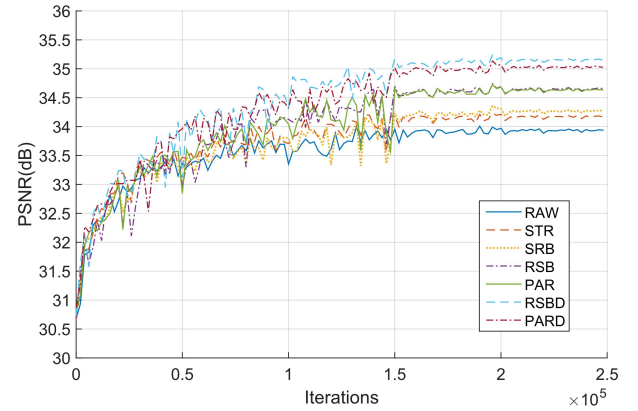


Fig. 13. The training performance of the seven networks. We drop the learning rate from 0.001 to 0.0001 when reaching 1.5×10^5 iterations and from 0.0001 to 0.00001 when reaching 2×10^5 iterations.

We compare the training performance and objective quality of these five versions on Rain20L with PSNR and SSIM as the evaluation metrics as shown in Figure 13 and 14 as well as in Table 5. The experimental results clearly show the superiority of **PAR** and **RSB**.

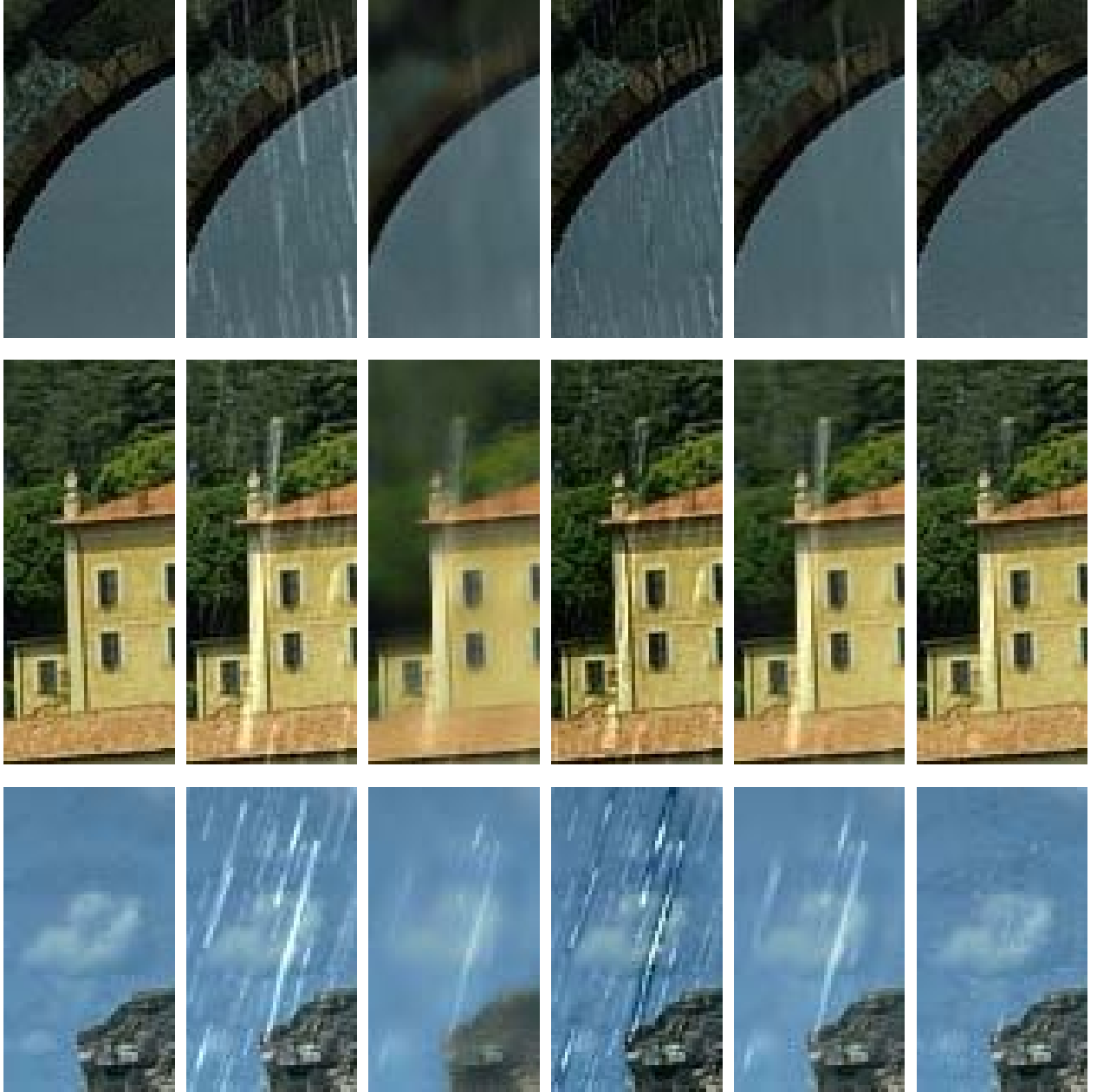


Fig. 8. Subjective results of different methods on Rain12. From left to right: Ground truth, rain image, ID, DSC, LP and JORDER. From top to bottom: *Image1*, *Image5* and *Image8*.

TABLE 4

The time complexity (s) of JORDER compared with state-of-the-art methods. *Jl* and *JlD* denote JORDER-IIF and JORDER-IIF-DEHAZE, respectively.

Scale	CNN (GPU)	ID	DSC	LP	JORDER (CPU)	JORDER (GPU)	Jl (GPU)	JlD (GPU)
80*80	0.85	449.94	14.32	35.97	2.97	0.11	0.32	0.72
500*500	6.39	1529.85	611.91	2708.20	69.79	1.46	3.08	7.16

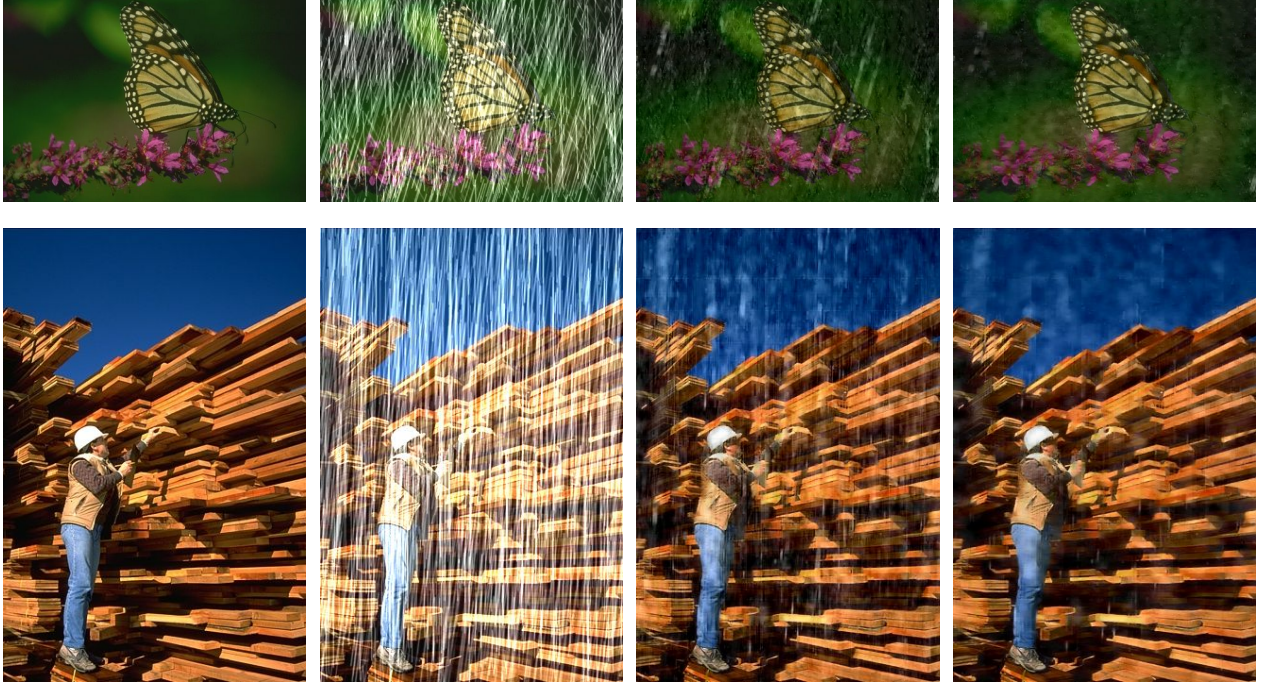


Fig. 9. Subjective results of different methods on Rain12. From left to right: Ground truth, rain image, JORDER, JORDER-IIF.

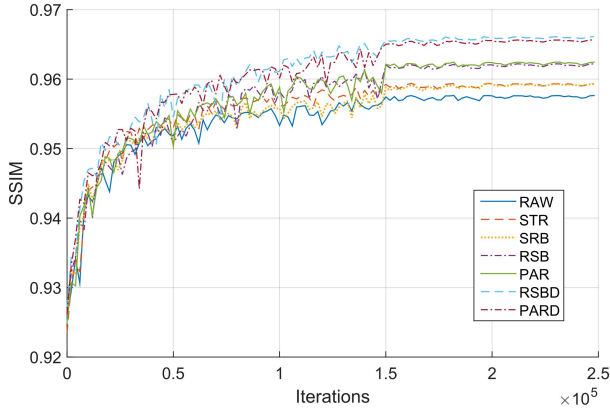


Fig. 14. The training performance of the seven networks. We drop the learning rate from 0.001 to 0.0001 when reaching 1.5×10^5 iterations.

TABLE 5
PSNR and SSIM results of the five versions.

Version	RAW	STREAK	PAR	RSB	SRB
PSNR	33.94	34.17	34.63	34.66	34.28
SSIM	0.9576	0.9593	0.9624	0.9622	0.9593

7.2 Analysis for Multi-Scale Aggreataion

We look into the benefit of multi-scale aggregation to the final performance. Three additional versions are involved in the comparisons: 6) **PARD**, boosted **PAR** with multi-scale aggregation; 7) **RSBD**, boosted **RSB** with multi-scale aggregation. 8) **JORDER-**, the version of JORDER without multi-scale aggregation. The training performance is also showed in Figure 13 and 14. The comparison for objective quality is shown in Table 6. The experimental results clearly demonstrate the positive effect of the multi-scale aggregation on the final objective performance.

TABLE 6
Objective evaluation for the effect of multi-scale aggregation.

Metric	PAR	PARD	RSB	RSBD	JORDER-	JORDER
Dataset	Rain20L		Rain20L		Rain12L	
PSNR	34.63	35.06	34.66	35.16	35.77	36.02
SSIM	0.9624	0.9655	0.9622	0.966	0.9412	0.9634

7.3 Order of Joint Derain and Dehaze

Figure 15 briefly demonstrates the results of joint derain and dehaze with an intermediate dehazing network embedded in different orders. As observed, a preprocessing dehazing (in Figure 15 (d)) makes weak rain streaks more obvious, some of which cause the failure of a successive deraining. A successive two deraining operations (in Figure 15 (c)) succeed in removing the rain streaks in high contrast regions. However, the atmospheric veiling effect still exists and the rain streaks covered by the veils are still preserved. A three-step operations – derain-dehaze-derain – first remove obvious rain streaks and noises, and then enhances the contrast and visibility, finally go on removing the enhanced rain streaks, as well as the artifacts caused by dehazing, achieving better visual quality.

8 CONCLUSION AND FUTURE WORKS

In this paper, we have introduced a new deep learning based method to effectively remove rain from a single image, even in the presence of rain accumulation and heavy rain. A new proposed region-dependent rain image formation model is proposed for additional rain detection and is further extended to simulate rain accumulation and heavy rains. Based on this model, we developed a new fully convolutional network that jointly detect and remove rain. Rain regions are first detected by the network which naturally provides additional information for rain removal. To restore images captured in the environment with both rain accumulation

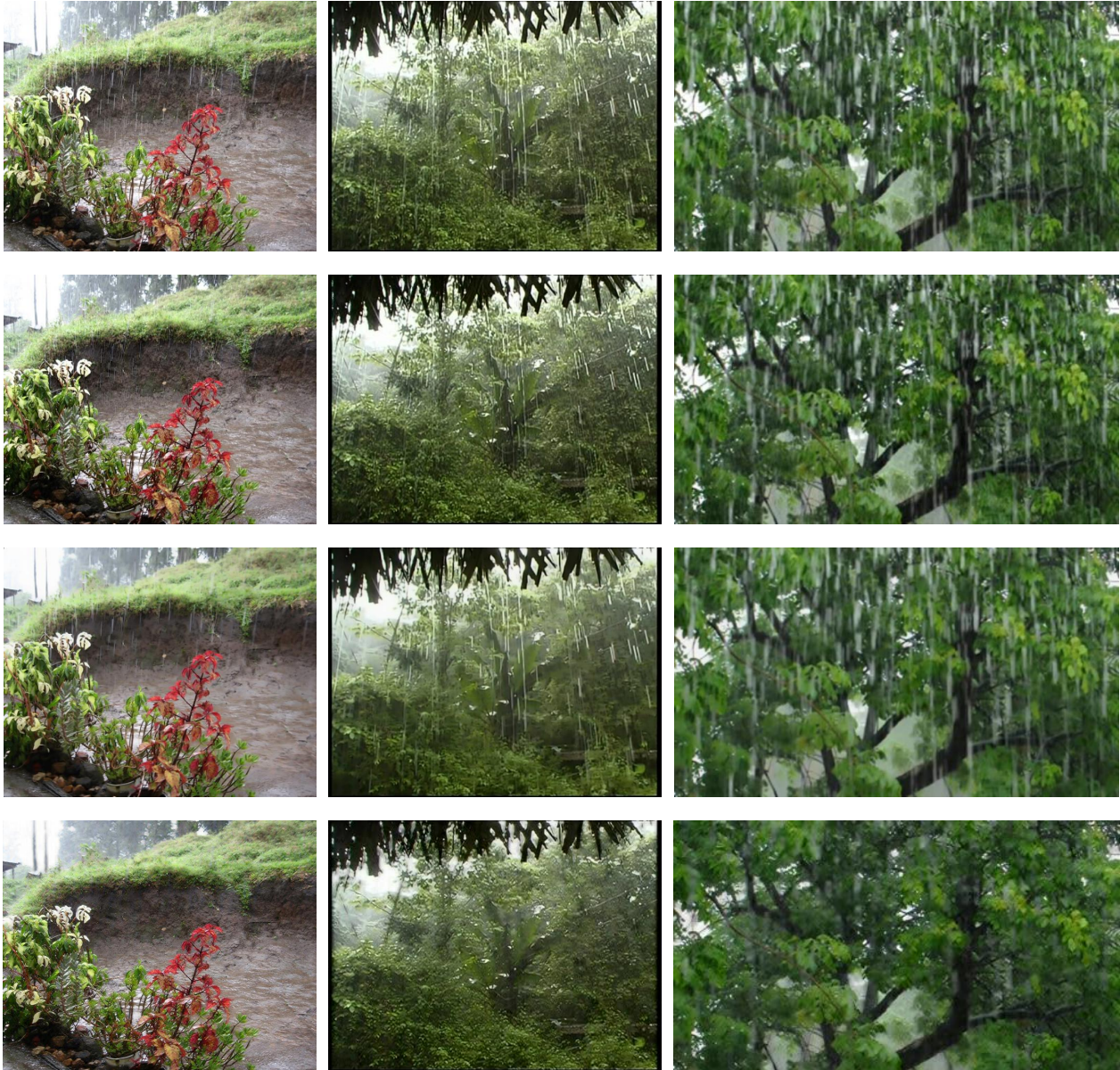


Fig. 10. Subjective results of different methods on real images. From top to down: rain image, DSC, LP and JORDER-IIF.



Fig. 11. The examples of JORDER-IIF-DEHAZE on heavy rain (the left two images) and mist images (the right two images).



Fig. 15. Results of joint derain and dehaze in different orders.

and heavy rain, we proposed an iterative information feedback network that progressively removes rain streaks, embedded with a dehazing network to remove atmospheric veils. Extensive evaluations demonstrated that our method outperforms state-of-the-art methods significantly on both synthesized data and real data.

REFERENCES

- [1] F. Agostinelli, M. R. Anderson, and H. Lee. Adaptive multi-column deep neural networks with application to robust image denoising. In *NIPS*. 2013.
- [2] P. C. Barnum, S. Narasimhan, and T. Kanade. Analysis of rain and snow in frequency space. *International Journal of Computer Vision*, 86(2-3):256–274, 2010.
- [3] J. Bossu, N. Hautière, and J.-P. Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International journal of computer vision*, 93(3):348–367, 2011.
- [4] N. Brewer and N. Liu. Using the shape characteristics of rain to identify and remove rain from video. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 451–458. Springer, 2008.
- [5] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising with multi-layer perceptrons, part 1: comparison with existing algorithms and with bounds. *arXiv:1211.1544*.
- [6] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising with multi-layer perceptrons, part 2: training trade-offs and analysis of their mechanisms. *arXiv:1211.1552*.
- [7] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *arXiv preprint arXiv:1601.07661*, 2016.
- [8] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, PP(99):1–1, 2016.
- [9] Y.-L. Chen and C.-T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1968–1975, 2013.
- [10] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen. Deep network cascade for image super-resolution. In *ECCV*. 2014.
- [11] C. Dong, C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2015.
- [12] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. In *ECCV*. 2014.
- [13] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 633–640, 2013.
- [14] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proc. IEEE Int'l Conf. Computer Vision*, December 2013.
- [15] K. Garg and S. K. Nayar. Detection and removal of rain from videos. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages 1–528. IEEE, 2004.
- [16] K. Garg and S. K. Nayar. When does a camera see rain? In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1067–1074. IEEE, 2005.
- [17] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 996–1002, New York, NY, USA, 2006. ACM.
- [18] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 996–1002. ACM, 2006.
- [19] K. Garg and S. K. Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007.
- [20] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv:1508.06576*, 2015.
- [21] D.-A. Huang, L.-W. Kang, Y.-C. F. Wang, and C.-W. Lin. Self-learning based image decomposition with applications to single image denoising. *IEEE Transactions on multimedia*, 16(1):83–93, 2014.
- [22] D.-A. Huang, L.-W. Kang, M.-C. Yang, C.-W. Lin, and Y.-C. F. Wang. Context-aware single image rain removal. In *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo, ICME '12*, pages 164–169. IEEE Computer Society, 2012.
- [23] D.-A. Huang, L.-W. Kang, M.-C. Yang, C.-W. Lin, and Y.-C. F. Wang. Context-aware single image rain removal. In *IEEE International Conference on Multimedia and Expo*, pages 164–169, Washington, DC, USA, 2012. IEEE Computer Society.
- [24] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.
- [25] V. Jain and S. Seung. Natural image denoising with convolutional networks. In *NIPS*. 2009.
- [26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, pages 675–678, New York, NY, USA, 2014. ACM.
- [27] L. W. Kang, C. W. Lin, and Y. H. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Trans. on Image Processing*, 21(4):1742–1755, April 2012.
- [28] J. H. Kim, C. Lee, J. Y. Sim, and C. S. Kim. In *Proc. IEEE Int'l Conf. Image Processing*.
- [29] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim. Single-image deraining using an adaptive nonlocal means filter. In *2013 IEEE International Conference on Image Processing*, pages 914–917. IEEE, 2013.
- [30] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 2736–2744, 2016.
- [31] C. Liu, Y. Pang, J. Wang, A. Yang, and J. Pan. Frequency domain directional filtering based rain streaks removal from a single color image.

- In *International Conference on Intelligent Computing*, pages 415–424. Springer, 2014.
- [32] K. G. Lore, A. Akintayo, and S. Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement. *arXiv preprint arXiv:1511.03995*, 2015.
 - [33] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 3397–3405, 2015.
 - [34] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.
 - [35] C. Osendorfer, H. Soyer, and P. van der Smagt. Image super-resolution with fast approximate convolutional sparse coding. In *Neural Information Processing*, 2014.
 - [36] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *International journal of computer vision*, 76(1):53–69, 2008.
 - [37] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Learning to deblur. *arXiv:1406.7444*, 2014.
 - [38] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang. Exploiting image structural similarity for single image rain removal. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4482–4486. IEEE, 2014.
 - [39] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 2010.
 - [40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
 - [41] H. Z. J. P. X. C. Wenqi Ren, Si Liu and M.-H. Yang. In *European Conference on Computer Vision*.
 - [42] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *NIPS*. 2012.
 - [43] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *NIPS*. 2014.
 - [44] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu. Automatic photo adjustment using deep neural networks. *TOG*, 2015.
 - [45] W. Yang, J. Feng, J. Yang, F. Zhao, J. Liu, Z. Guo, and S. Yan. Deep Edge Guided Recurrent Residual Learning for Image Super-Resolution. *ArXiv*, April 2016.
 - [46] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.
 - [47] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *ArXiv e-prints*, August 2016.
 - [48] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng. Rain removal in video by combining temporal and chromatic properties. In *2006 IEEE International Conference on Multimedia and Expo*, pages 461–464. IEEE, 2006.